Upload Data Rate for Optimal First Chunk Download Time for P2P based Video on Demand (VoD) Services

Sudipta Majumder
Dept. of CSE, DUIET,
Dibrugarh University,
Assam, India
Email:sudipta2020@gmail.com

Md. Anwar Hussain
Dept. of ECE
NERIST, Nirjuli,
Arunachal Pradesh, India
Email:bubuli_99@yahoo.com

Abstract- Video on demand (VoD) systems are the one of the emerging content distribution mechanism as it adds lots of conveniences and choices to the users where they can watch Video as per their will asynchronously. But, the implementation of such system faces lots of challenges such as wait-time for starting of the service, download time of the whole video content to provide continuous & unhindered services etc. In this paper we have studied the relation between download time of first chunk and the whole video if the bandwidth is limited in perspective of upload rate in P2P based Video on demand system. Here, we find out at what Upload data rate, we can get best of trade-offs between first chunk download time and whole video download time in order to continuous and unhindered services.

Keywords- *Video on demand (VoD), Peers, trackers, Bittorrent, seeder, chocking, Upload data rate, chunk, Download time.*

*****

I. INTRODUCTION

Peer-to-peer (**P2P**) computing or networking is a distributed application architecture that partitions tasks or workloads between peers. Peers are equally entitled, equipotent participants in the application. These nodes are assumed to form a peer-to-peer network of nodes.

Peers make a portion of their resources, such as processing power, computing power or network bandwidth, directly available to other network participants, without the need for central administration by servers or stable hosts. Peers are both servers and consumers of resources, in comparison to the traditional client-server model in which the consumption and supply of resources is divided. In this paper, we are trying to understand the relation between Upload capacity and numbers of seeds with start time, First chunk download time and total download time. For this we have created closed simulation environment

Scenariohaving 10 peers participating in it and size of Video chunk considered is 100MB.

Peer-to-peer file sharing became popular in 1999 with the introduction of Napster, a file sharing application and a set of central servers that linked people who had files with those who requested files. The central index server indexed the users and their shared content. When a node is searching for a particular segment of the file, the server searches all available copies of that file and return a list of them to the searching node. Here the transfer of the searched file takes place between the two private computers. A limitation was that only music files could be shared [2]. Because this process occurred on a central server, however, Napster was held liable for copyright infringement and shut down in July 2001. It later reopened as a pay service [1].After Napster was shut down, the most popular peer-to-peer services were Gnutella and Kazaa. These services also allowed users to download files other than music, such as videos, movies and games.

Some of the successful peer to peer based content delivery systems are Gnutella [4],KaZaA [5], eDonkey [3], and BitTorrent [10]. These p2p systems are found to be very popular among the user. These systems are generally used to distribute media files such as video files. But the disadvantage of the systems is that the whole video has to be downloaded before the user start watching them. To overcome such a disadvantage Cool streaming [13] and its derivate ([4], [6]) have come up with system that can be used to stream live media contents. But the question of recent years is that if peer to peer based systems can be used to provide a near-Video-on-Demand (VoD) service to the users.

A near-Video-on-Demand (VoD) has got a vital advantage which is that the video can be watched while it's being downloaded and besides this, another advantage that it can be watched whenever we want after it has been downloaded and operations like rewind and fast-forward on the video file makes it a very desirable and useful service.

In this paper, we investigate the potential of leveraging P2P networks for a VoD service and provide guidance, and designs principles to efficiently build such systems.Video distribution over the Internet has been one of the most prolific areas of research [8], [9], [15], [22]. The particular problem of designing a near-VoD service has also received extensive attention in the past [7], [11], [14], [15], [16]. An important requirement of a near-VoD service is to be able to support a large number of users, and, hence, such systems should be scalable. The need for scalability becomes clear if we consider that a typical video stream incurs a heavy burden both on the network and the system resources (e.g. disk I/O) of the server. The multicasting paradigm has been proposed to address the scalability issues [7], [12], [15]. Indeed, many systems such as Pyramid Broadcasting [15] and Skyscraper [12] can provide scalable Near-VoD service by using elegant techniques for dividing the video into segments and broadcasting each segment in a different multicast channel. However, both these systems require a multicast-enabled infrastructure.

II. P2P VIDEO ON DEMAND

The Main Advantages of Video-on-demand (VoD)service systems provide multimedia services offer more flexibility and convenience to users by allowing them to watch any kind of video at any point in time. Such Video-on-demand (VoD)systems are made in such a way that capable of delivering the requested Video cautiously to the user. Unlike live streaming, there is complete control over the media,in VoD systems, with operations such as pause, forward and backward functionalities. The desired behavior of VoD systems is that it can handle large number of video demands made by users asynchronously for watching different parts of the same video at any given time. Implementation of VoD system having such behavior is very challenging specially in tree- based p2p systems because the users will get video exactly in the same order it left the root node as they receive contents directly from the source server. Butthe Mesh-based P2P systems can handle such situation efficiently as they are able to distribute large files. In this kind of systems a large video is usually broken into many small blocks of pieces. Both the system throughput and the rate, at which the content can be distributed to users, greatly depend on the diversity of the blocks contained at different peers. The challenge of providing VoD services using mesh-based P2P networks lies in the fact that the blocks have to be received at the peer-side in a sequential order, and time constraints have to be considered at all times to guarantee continuous visualization. Hence VoD services using mesh-based P2P can effectively address all the issues related to the efficient Video-on-demand (VoD) services. In the next sub- section we are going to discuss tree based P2P based VoD and Mesh Based VoD to little extent.

A. Tree-based P2P VoD

One of the first IP multicast policies introduced for supporting VoD services waspatching [24]. It inspired P2Cast [25] design for distributing video content among asynchronous users. Here each user act as a server for the video content that it has, while receiving portions of videos from other users. Users joining the P2P network within certain time limit formpart of a session. Along with the source server, users belonging to the same session form anALM tree, known as the base tree. Then, the entire video is streamed from the source server using this base tree. Users joining the session join the base tree and retrieve the base stream from it. In addition, new clientsmissing the initial part of the video must obtaina get the video directly from theconcerned server or other users whohave already cached the required video. Users behave just like peers in a P2P network.

B. Mesh-based P2P-VoD

In a mesh overlay network, eachnode contacts a subset of neighbors to receive a number ofchunks. Each node needs to know which chunks are ownedby its neighbors and explicitly pulls the chunks it needs. Inessence, every node relies on multiple neighbors to retrieve thecontent which makes the system resilient to node-failures. It isworth noting that a peer may simultaneously request multipleneighbors in two ways, namely for either the same contentor for different ones [12]. The Users should receive blocks sequentially in order to watch the movie while downloading[19]. Additionally, the nature of VoD systems should ensure the availability of different blocksof video file at any given time, especially if users expect to perform VCR operations during playback [20].

III. IMPLEMENTING OF SIMULATIONSCENARIOS

We have simulated the scenarios for p2p based BitTorrent Video on demand in NS2. NS 2 is a discrete event simulator. The scalability and robustness of the simulator have attracted us for choosing the video on demand simulation platform.Followings are the list of BitTorrent VoD parameters are used for simulation.

```
// inform TCL script when peer finishes download
#define BT_TELL_TCL

// choking algorithm used in the simulation
#define BITTORRENT 0
#define BITTORRENT_WITH_PURE_RAREST_FIRST 1

// rolling average (multiple of CHOKING_INTERVAL)
// In BT: 20s
#define ROLLING_AVERAGE                  2

// time interval between two optimistic unchokes (multiple
of CHOKING_INTERVAL)
// In BT: 30s
#define OPTIMISTIC_UNCHOKE_INTERVAL      3

// time interval between remote peer has to upload at least
one piece to be not snubbed
// In BT: 60s
#define ANTI_SNUBBING                    6

// number of missing pieces to switch from normal mode to
end-game
#define END_GAME                 1

// time to wait before requesting more peers (default = 300)
#define REREQUEST_INTERVAL 300

// time to wait between checking if any connections have
timed out (defaults to 300.0)
#define TIMEOUT_CHECK_INTERVAL   300
```

// number of seconds to pause between sending keep alives (defaults to 120.0)
#define KEEPALIVE_INTERVAL 120
The test bed for the simulation includes user entered number of peers. Besides this, the test bed allows us desired number of seeders. The upload rate for all the peers can be determined at the run time. Here all the peers participating in the simulation form a neighborhood. A peer can download the desired block from the neighbor if it has the block or it shares the video. Here a tracker is used to keep track of all the peers and the blocks of video file

A Simulation parameters

TABLE 1: SIMULATION SCENARIO 1,2,3

Parameter	Value
No of Peers (N_P)	10
No of Seeds (N_S)	1
File Size (S_F_MB)	100 MB
Chunk Size (S_C)	256 KB
No. of Chunks (N_C)	400
Choking Algorithm	Pure Rarest First
Keep alive interval	120 Sec.
Time out check interval	300 Sec
Peer Registered at tracker	10
Upload rate	500 Kbps,600 kbps 700 Kbps

TABLE 2 SIMULATION SCENARIO 4,5,6

Parameter	Value
No of Peers (N_P)	10
No of Seeds (N_S)	1
File Size (S_F_MB)	100 MB
Chunk Size (S_C)	256 KB
No. of Chunks (N_C)	400
Choking Algorithm	Pure Rarest First
Keep alive interval	120 Sec.
Time out check interval	300 Sec
Peer Registered at tracker	10
Upload rate	800 Kbps,900 kbps 1000 Kbps

TABLE 3: SIMULATION SCENARIO 7,8,9

Parameter	Value
No of Peers (N_P)	10
No of Seeds (N_S)	2
File Size (S_F_MB)	100 MB
Chunk Size (S_C)	256 KB
No. of Chunks (N_C)	400
Choking Algorithm	Pure Rarest First
Keep alive interval	120 Sec.
Time out check interval	300 Sec
Peer Registered at tracker	10
Upload rate	500 Kbps,600 kbps 700 Kbps

TABLE 4: SIMULATION SCENARIO 10, 11 and 12

Parameter	Value
No of Peers (N_P)	10
No of Seeds (N_S)	2
File Size (S_F_MB)	100 MB
Chunk Size (S_C)	256 KB
No. of Chunks (N_C)	400
Choking Algorithm	Pure Rarest First
Keep alive interval	120 Sec.
Time out check interval	300 Sec
Peer Registered at tracker	10
Upload rate	800 Kbps, 900 kbps 1000 Kbps

TABLE 5: SIMULATION SCENARIO 13, 14 and 15

Parameter	Value
No of Peers (N_P)	10
No of Seeds (N_S)	3
File Size (S_F_MB)	100 MB
Chunk Size (S_C)	256 KB
No. of Chunks (N_C)	400
Choking Algorithm	Pure Rarest First
Keep alive interval	120 Sec.
Time out check interval	300 Sec
Peer Registered at tracker	10
Upload rate	500 Kbps, 600 kbps 700 Kbps

TABLE 6: SIMULATION SCENARIO 16, 17and 18

Parameter	Value
No of Peers (N_P)	10
No of Seeds (N_S)	3
File Size (S_F_MB)	100 MB
Chunk Size (S_C)	256 KB
No. of Chunks (N_C)	400
Choking Algorithm	Pure Rarest First
Keep alive interval	120 Sec.
Time out check interval	300 Sec
Peer Registered at tracker	10
Upload rate	800 Kbps, 900 kbps 1000 Kbps

 ISSN: 2321-8169
2967 - 2972

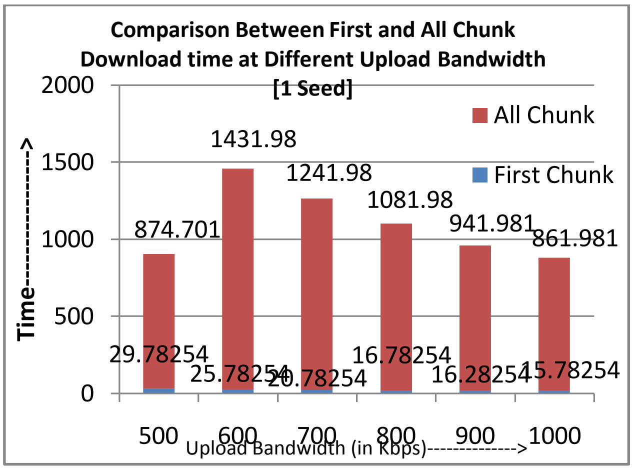

Figure 1: Comparison between First and All Chunk Download time at Different Upload Bandwidth [1 Seed]

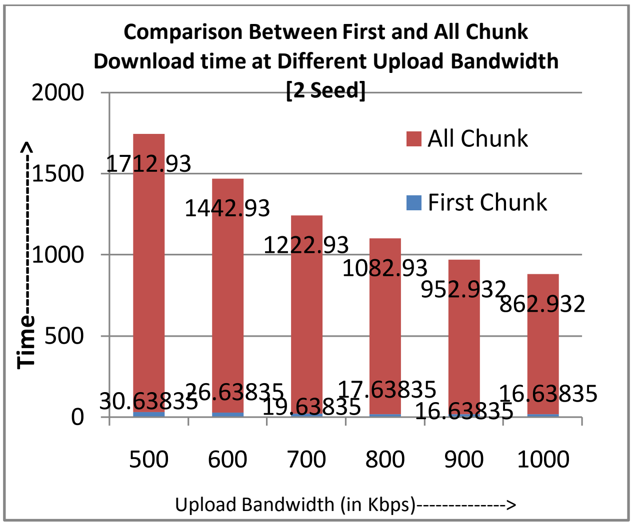

Figure 2: Comparison between First and All Chunk Download time at Different Upload Bandwidth [2 Seed]

In figure 1, 2 and 3 we have studied the relation among of Seed,upload bandwidth and time taken to download the first chunk and all the chunk required to view the Video which is of 100 Mb. Down load time of the first chunk is important in p2p based video on demand services because it determines the wait time of the viewing the video or media. The wait time is critical quantity because if it is very high, it will affect the consumer as nobody has much time waiting for a video nowadays until and unless it is very urgent. The rest of the chunk of the video will be downloaded while the consumer is watching the first chunk. They will not require to wait for next chunk to be downloaded.

It is found by studying the data and the graphs that the waiting time for the first chunk decreases as the upload bandwidth increases. It is evident from the figure 1, 2 and 3. Besides this the percentage of time consumed for downloading the first chunk in respect of downloading the whole video firstdecreases then increases. It is clear from the following tables 7, 8 and 9.

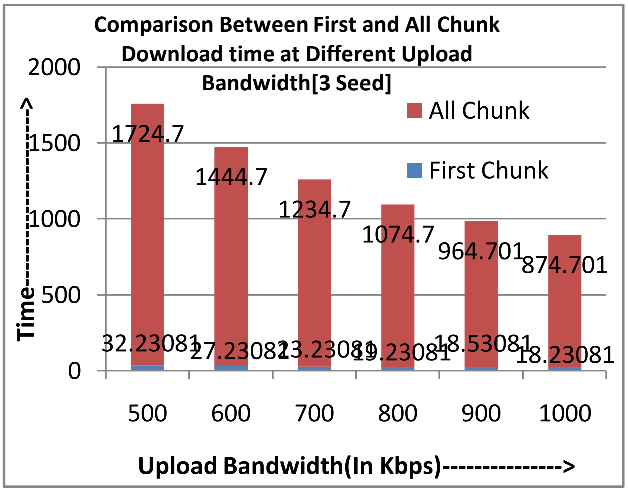

Figure 3 Comparison between First and All Chunk Download time at Different Upload Bandwidth [3 Seed]

Table 7: Percentage time of first chunk to download in 1 seed

	500 kbps	600 kbps	700 kbps	800 kbps	900 kbps	1000 kbps
Fist Chunk	29.78254	25.78254	20.78254	16.78254	16.28254	15.78254
All Chunk	874.701	1431.98	1241.98	1081.98	941.981	861.981
% Time	3.40	1.80	1.67	1.55	1.73	1.83

Table 8: Percentage time of first chunk to download in 2 seed

	500kbps	600kbps	700kbps	800kbps	900kbps	1000kbps
Fist Chunk	30.63835	26.63835	19.63835	17.63835	16.63835	16.63835
All Chunk	1712.93	1442.93	1222.93	1082.93	952.932	862.932
%Time	1.79	1.85	1.61	1.63	1.75	1.93

Table 9: Percentage time of first chunk to download in 3 seed

	500kbps	600kbps	700kbps	800kbps	900kbps	1000kbps
Fist Chunk	32.23081	27.23081	23.23081	19.23081	18.53081	18.23081
All Chunk	1724.7	1444.7	1234.7	1074.7	964.701	874.701
%Time	1.87	1.88	1.88	1.79	1.92	2.08

As we can see from the tables 7, 8 and 9, and from the figures 1, 2 and 3 that the percentage of time taken to download the first chunk of the video is least at upload data rate between 700 kbps to 800 kbps. This data set is chosen from hundreds of simulated scenarios with varying upload data rate. It is observed that for the given simulation scenario with varying upload data rates, the time taken to download each and every chunks is most likely to limited to a small range of time. That is, the time taken to down load the chunks won't vary very drastically which is one of the desired characteristics for a p2p based video on demand system. Also the figures 1 to 3 tells us that as the upload bandwidth of the peers are increased, the time required to download the entire chunks also decrease. But there have to be a tradeoff between upload bandwidth and download bandwidth otherwise too much increase in the upload bandwidth will spend the overall bandwidth resulting more waiting time for the first chunk.

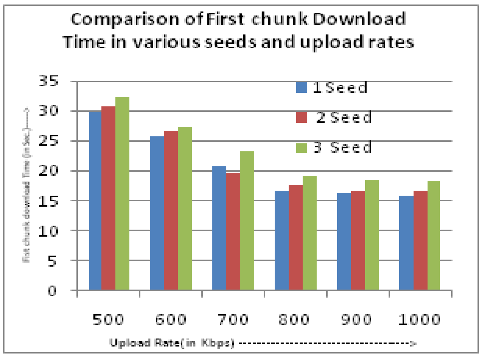

Figure 4 Comparison of First chunk Download Time in various seeds and upload rates

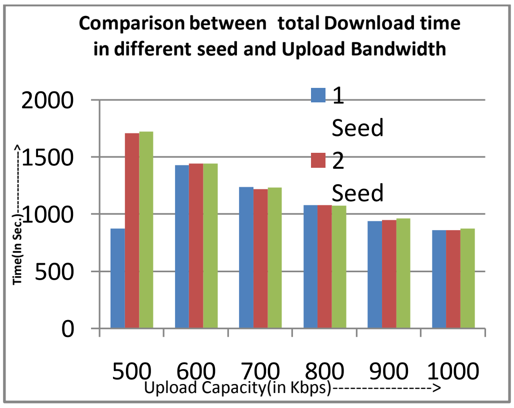

Figure 5Comparison between Different total Download time in different seed and Upload Bandwidth

A seed refers to a machine possessing some part of the data. A peer or downloader becomes a seed when it starts uploading the already downloaded content for other peers to download from. This includes any peer possessing 100% of the data or a web seed. When a downloader starts uploading content, the peer becomes a seed. Here in the simulation part, we have made three kinds of simulations. Some simulations are having only one seed, some having two numbers of seed and rest are having three numbers of seeds.

Seeding refers to leaving a peer's connection available for downloaders to download from. Normally, a peer should seed more data than download. However, whether to seed or not, or how much to seed, depends on the availability of downloaders and the choice of the peer at the seeding end. In figure 4, we can see that the time taken to download the first chunk is small in simulations having only one seed in compared to simulations more numbers of seed. The time taken shown in the figures is average.

IV. CONCLUSION

In this paper, we have deduced a relation between upload and download time of videos in p2p based Video on demand system in respect of first chunk of video to be downloaded in whole video download. If the download time of the first chunk is more, then the viewers will have to wait more to watch the video. Whereas if the total-download time of the whole video is more, then it may effect continuous viewing of it. So there must be tradeoff between download time of first chunk and the whole video. This trade-off is balanced by the upload and download data rate of peers in the p2p based Video on demand.

From the figures above, we find that the first chunk download time decreases with the increase of upload data rate. Similarly the download time of the whole videos are also decreasing with the increase of upload capacity as shown in the tables. But here, the rate of decreasing time to download for first chunk and whole video if different. The percentage of time taken to download the first chunk with respect to whole video first decreased and then increased which is clear from the table 7,8 & 9. On careful study of the graphs and table, we find that upload data rate of 700kbps to 800kbps are optimal for small P2Pbased Video on demandnetwork with limited bandwidth.With upload data rate of 700kbps to 800kbps, users/viewers won't have to wait much for the first chunk and continue watching the videos with minimum or no hindrance due download time.

REFERENCE:

[1] Carmack,Carman "How Bit Torrent Works".
[2] Tyson, Jeff.. "How the Old Napster Worked".
[3] eDonkey2000 - overnet. http://www.edonkey2000.com/.
[4] Gnutella.p://p2pjournal.com/main/gnutella.htm.
[5] KaZaA. http://www.kazaa.com/.
[6] Pplive. http://www.pplive.com/.
[7] Kevin C. Almeroth and Mostafa H. Ammar. On the use of multicast delivery to provide a scalable and interactive Video-on-Demand service. Journal of Selected Areas in Communications,14(6):1110–1122, 1996.
[8] John G. Apostolopoulos, Waitian Tan, and Susie J.Wee. Video streaming: Concepts, algorithms, and systems.http://www.hpl.hp.com/techreports/2002/HPL-2002-260.pdf, Sep 2002.
[9] Shanwei Cen, Calton Pu, Richard Staehli, Crispin Cowan, and Jonathan Walpole. A distributed real time MPEG video audio player. In NOSSDAV, 1995.
[10] BramCohen. BitTorrent. http://www.bittorrent.com.
[11] Martin ReissleinDespinaSaparilla, Keith Ross. Periodic broad-casting with vbr-encoded video. In IEEE Infocom.IEEE Press, 1999.

ISSN: 2321-8169
2967 - 2972

[12] ien A. Hua and Simon Sheu. Skyscraper broadcasting: A new broadcasting scheme for metropolitan video-on-demand systems. In ACM SIGCOMM, pages 89–100. ACM Press, 1997.

[13] V.P. Kompella, J.C. Pasquale, and G.C Polyzos. Multicasting for multimedia applications. In IEEE Infocom'92, volume 3, pages 2078–2085. IEEE Press, May 1992.

[14] Sheu, Kien Hua, and WallapakTavanapog. Chaining: A generalized batching technique for video-on-demand systems. In International Conference on Multimedia Computing and Systems (ICMCS'97). IEEE Press, 1997.

[15] S. Viswanathan and T. Imiehnski. Pyramid broadcasting for video on demand service. In IEEE Multimedia Computing and Networking Conference. IEEE Press, 1995.

[16] X. Zhang, J. Liu, B. Li, and T.-S. P. Yum. CoolStreaming/DONet: A data-driven overlay network for peer-to-peer live media stream-ing. In IEEE Infocom. IEEE Press, 2005

[17] EMULE. Emule Homepage. http://www.emule-project.net/.

[18] Guo, Y., Suh, K., Kurose, J., &Towsley, D. (2003). "P2Cast: Peer-to-peer patching scheme for VoD service". Proceedings of the 12th International Conference on World Wide Web, 301-309.

[19] Annapureddy, S., Guha, S., Gkantsidis,C., Gunawardena, D., &Rodriguez,P. R. (2007). Is high-quality vod feasible using P2P swarming?Proceedings of the 16th International Conference on World Wide Web, 903-912.

[20] Hareesh.K and Manjaiah D.H " Peer-to-peer live streaming and video on demand design issues and its challenges", IJP2P,Vol.2, No.4, October 2011

[21] Kolja Eger, Tobias Hoßfeld, Andreas Binzenhofer, Gerald Kunzmann"Efficient Simulation of Large-Scale P2P Networks:Packet-level vs. Flow-level Simulations"Proceeding UPGRADE '07 Pages 9 - 16 ACM New York, NY,USA.

[22] A. Betker, C. Gerlach, R. H¨ulsermann, M. J¨ager,M. Barry, S. Bodamer, J. Sp¨ath, C. Gauger, andM. K¨ohn.Reference transport network scenarios.MultiTeraNet Report, 2003.